Disentangling 3D/4D Facial Affect Recognition With Faster Multi-View Transformer

Muzammil Behzad, Xiaobai Li, *Member, IEEE*, and Guoying Zhao, *Senior Member, IEEE*

***Abstract*—In this paper, we propose MiT: a novel multi-view transformer model[1] for 3D/4D facial affect recognition. MiT incorporates patch and position embeddings from various patches of multi-views and uses them for learning various facial muscle movements to showcase an effective recognition performance. We also propose a multi-view loss function that is not only gradient-friendly, and hence speeds up the gradient computation during back-propagation, but it also leverages the correlation associated with the underlying facial patterns among multi-views. Additionally, we offer multi-view weights that are trainable and learnable, and help substantially in training. Finally, we equip our model with distributed performance for faster learning and computational convenience. With the help of extensive experiments, we show that our model outperform the existing methods on widely-used datasets for 3D/4D FER.**

***Index Terms*—Affect, emotion recognition, multi-views, transformer, 3D/4D faces.**

I. Introduction

ATTENTION-BASED architectures, specifically Transformers [1], have shown exemplary performance in natural language processing (NLP) tasks. One of the main ingredients of Transformer's success is their ability to be pre-trained on large data, and then having the capability of fine-tuning for specific tasks [2], [3]. Equipped with computational efficiency and scalability, and with no sign of saturating performance [4], the tremendous breakthroughs shown by Transformers have ignited eager interest in the computer vision community [5] to adapt these models for vision-related tasks [6]–[11].

Inspired by the scaling victory of Transformers, we work towards developing a transformer architecture to disentangle the correlated patterns in 3D/4D faces for effective facial expression recognition (FER). Specifically, contrary to the 2D faces (e.g., [12]–[15]), such expression recognition involves predicting emotions from 3D/4D faces with complementary spatial and temporal facial features, and the significant results [16]–[19] have proven its merits.

To learn from the underlying 3D facial geometry, several methods exist in the literature. Generally, the most popular approaches can be categorized as local feature-based [19]–[21], template-based [22]–[24], curve-based [25], [26] and 2D projections-based [27], [28] approaches. Recently, 4D FER has received great deal of interest for allowing deep learning models to learn effective facial cues. For example, Yin *et al.* [29] and Sun *et al.* [30] used Hidden Markov Models (HMM) to train the network with temporal facial features via 4D faces. Likewise, by using the random forest classifier, Ben Amor *et al.* [31] demonstrated that a deformation vector field based on Riemannian analysis could produce handy results. Similarly, Sandbach *et al.* [32] used HMM and GentleBoost for classifying the proposed free-form representations of the 3D frames. Additionally, the authors in [33] represented geometrical coordinates and its normal as feature vectors, and as dynamic local binary patterns (LBP) in another work [34] for classifying expressions with support vector machine (SVM). In a similar way, the authors in [35] extract features from polar angles and curvatures, and propose a spatio-temporal LBP-based feature extractor for classification. Although these works report desirable results, the use of local and manually extracted features make the solutions practically inconvenient.

In contrast, Li *et al.* [36] proposed an interesting model for automatic 4D FER via dynamic geometrical image network. They generate geometrical images by estimating the differential quantities from the given facial point clouds. The final emotion prediction is then a result of score-level fusion from the probability scores of different geometrical images. Another recent method takes into account the sparse coding-based representations of LBP difference [37]. In this work, the appearance and geometric features are first extracted via mesh-local binary pattern difference (mesh-LBPD), and then sparse coding is applied to predict emotions.

Recently, some works [38], [39] used attention-based models or Transformers for 2D FER and they do not perform 3D/4D FER. Nevertheless, for a robust FER, an ideal algorithm should look beyond the apparent feature representations. For example, the role of multi-views is often ignored its ability to leverage effective facial cues are not explored.

A. Contributions

In the light of above discussion, we highlight our contributions here. As a backbone, we use Vision Transformer (ViT) [4]

Manuscript received August 22, 2021; revised September 3, 2021; accepted September 3, 2021. Date of publication September 9, 2021; date of current version September 30, 2021. This work was supported in part by Infotech Oulu and the Academy of Finland through project MiGA under Grant 316765, project 6 + E under Grant 323287, and ICT 2023 project under Grant 328115; in part by the Riitta ja Jorma J. Takanen Foundation; and in part by the Tauno Tönning Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Victor Sanchez. *(Corresponding author: Guoying Zhao.)*

The authors are with the Center for Machine Vision and Signal Analysis, University of Oulu, Oulu 90570, Finland (e-mail: muzammil.behzad@oulu.fi; xiaobai.li@oulu.fi; guoying.zhao@oulu.fi).

Digital Object Identifier 10.1109/LSP.2021.3111576

[1]https://github.com/muzammilbehzad/MultiviewTransformer

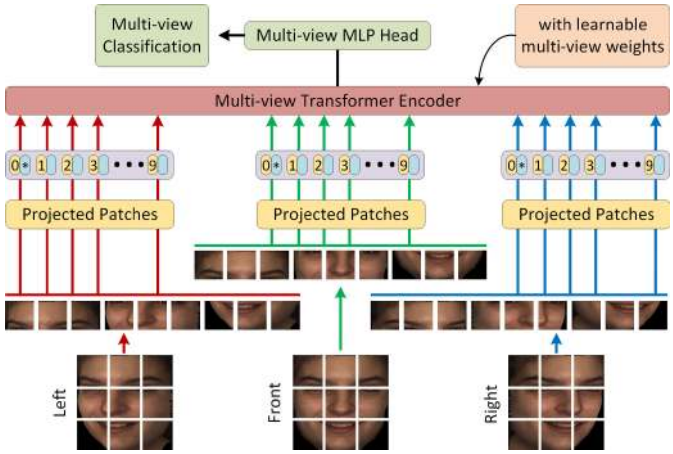

Fig. 1. Multi-view Transformer's overview: we split multi-view faces into multi-view patches of fixed-size, linearly embed each of them, add multi-view position embeddings, and feed the resulting sequence of vectors to our proposed multi-view Transformer encoder. We add learnable classification token to the sequence for each view, and for classification, we add extra learnable multi-view weights.

which applies the original Transformer [1]. However, we substantially extend this backbone to tailor our proposed multi-view transformer architecture. In particular, our contributions are as following:

1) We propose a novel multi-view transformer (MiT) that incorporates patch and position embedding from various patches of multi-views.
2) Instead of using independent losses, we propose a multi-view loss function to facilitate gradient updates across multi-views during backward propagation.
3) Rather than using fixed/manual weights, we propose multi-view weights that are trainable and learnable.
4) For computational and practical convenience, we equip our code with faster distributed training performance[2].

Finally, to our best knowledge, MiT is the pioneer multi-view transformer architecture for 3D/4D facial data.

II. Our Proposed Model

Since our MiT model aims to incorporate multi-views, a benefit of this tremendously robust setup is that scalable and efficient NLP Transformer architectures – and their effective implementations – can be utilized almost out of the box.

A. Multi-View Transformer (MiT)

We present an overview of our model in Fig. 1. To handle multi-view 2D images, we reshape each image $\mathbf{X}_\theta \in \mathbb{R}^{H\times W\times C}$ into train a flattened patches $\mathbf{x}_{p,\theta} \in \mathbb{R}^{H\times(P^2.C)}$, where (H, W) denotes input image resolution of each view, θ is the rotation angle (20° for left, 0° for right, and -20° for front) for each view, C is the number of channels, (P, P) is the size of extracted patch, and $N = HW/P^2$ refers to total number of patches in each view. Since the transformer uses constant latent vector size κ in its layers, we map the flattened patches to κ dimensions

[2]https://github.com/muzammilbehzad/FasterDistributedTraining

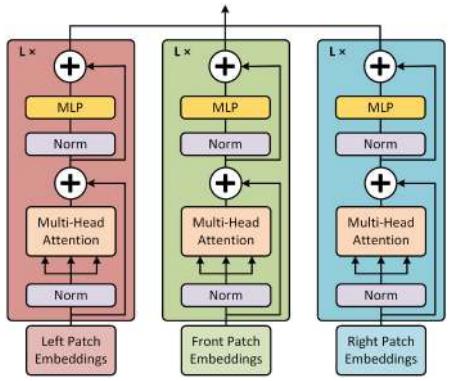

Fig. 2. Illustration of the proposed multi-view Transformer encoder.

with a trainable linear projection as,

$$\varepsilon_{0,\theta} = [\mathbf{x}_{class}; \mathbf{x}^1_{p,\theta}\mathbf{E}; \mathbf{x}^2_{p,\theta}\mathbf{E}; \ldots \mathbf{x}^N_{p,\theta}\mathbf{E};] + \mathbf{E}_{pos},$$
$$\mathbf{E} \in \mathbb{R}^{(P^2.C)\times\kappa}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times\kappa}, \quad (1)$$

where the output of this projection in (1) corresponds to patch embeddings for each θ. As a class token, we prepend learnable embedding to the sequence of embedded patches for each view ($\varepsilon^0_{0,\theta} = \mathbf{x}_{class}$) whose state at the Transformer's encoder output ($\varepsilon^0_{L,\theta}$) corresponds to the image representation $\mathbf{Y}_\theta$. To train positional knowledge, position embeddings are added to patches via standard learnable 1D position embeddings. Consequently, the sequences of embedding vectors for each view serve as input to the encoder. As shown in Fig. 2, the encoder contains alternating layers of multi-headed self-attention, and multilayer perceptron (MLP) blocks, where the residual connections are applied after every block, while layernorm is applied before every block as,

$$\varepsilon'_{l,\theta} = \eta(\Phi(\varepsilon_{l-1,\theta})) + \varepsilon_{l-1,\theta}, \quad \forall l, \quad (2)$$
$$\varepsilon_{l,\theta} = \mu(\Phi(\varepsilon'_{l,\theta})) + \varepsilon'_{l,\theta}, \quad \forall l, \quad (3)$$

where l is the layer and, $\eta(.)$, $\mu(.)$ and $\Phi(.)$ refer to multi-headed self-attention, MLP with two layers and GELU non-linearity, and layernorm operator, respectively, and the image representation for each view is defined by $\mathbf{Y}_\theta = \Phi(\varepsilon^0_{L,\theta})$.

B. Multi-View Loss

Our transformer model distinguishes itself from other models via its innovative loss function. Instead of computing independent losses for each facial view, we propose a multi-view loss function to leverage the loss associated with the underlying correlated facial patterns among multi-views. This collaborative approach not only yields effective performance results, but it is also gradient-friendly thereby allowing faster processing. We therefore formulate our multi-view loss as:

$$\mathcal{L}_\mathcal{M} \triangleq -\sum_i y_i \ln\left(\sum_\theta \omega_\theta \hat{y}_{i,\theta}\right), \quad (4)$$

where y_i is the label, and $\hat{y}_{i,\theta}$ is model's prediction at each view, and ω_θ are learnable weights learned while training. Importantly, we theoretically demonstrate the gradient-friendly nature of our proposed loss function advocating its efficiency for faster network training. Let φ_j denote the model's learnable parameter/weight at index j, we can then compute the derivative

of (4) w.r.t the learnable parameter as:

$$\frac{\partial \mathcal{L}_{\mathcal{M}}}{\partial \varphi_j} = -\frac{\partial}{\partial \varphi_j} \sum_i y_i \ln(\sum_\theta \omega_\theta \hat{y}_{i,\theta}), \tag{5}$$

$$\frac{\partial \mathcal{L}_{\mathcal{M}}}{\partial \varphi_j} = \underbrace{-\sum_{i \neq j} y_i \frac{1}{\sum_\theta \omega_\theta \hat{y}_{i,\theta}} \times \frac{\partial}{\partial \varphi_j}(\sum_\theta \omega_\theta \hat{y}_{i,\theta})}_{i \neq j}$$

$$\underbrace{- y_j \frac{1}{\sum_\theta \omega_\theta \hat{y}_{j,\theta}} \times \frac{\partial}{\partial \varphi_j}(\sum_\theta \omega_\theta \hat{y}_{i,\theta})}_{i=j} \tag{6}$$

Using the standard derivation of the gradients, we express the derivative terms in (6) as,

$$\frac{\partial}{\partial \varphi_j} \sum_\theta \omega_\theta \hat{y}_{i,\theta} = \begin{cases} \sum_\theta \omega_\theta \hat{y}_{i,\theta}(1 - \sum_\theta \omega_\theta \hat{y}_{i,\theta}), & i = j \\ -\sum_\theta \omega_\theta \hat{y}_{i,\theta} \sum_\theta \omega_\theta \hat{y}_{j,\theta}, & i \neq j \end{cases} \tag{7}$$

Manipulating and rearranging the terms in (6) and (7) yields

$$\frac{\partial \mathcal{L}_{\mathcal{M}}}{\partial \varphi_j} = \sum_\theta \omega_\theta \hat{y}_{j,\theta} - y_i, \tag{8}$$

which translates into a very elegant and gradient-friendly expression especially for computational reasons.

C. Trainable Multi-View Weights

To effectively incorporate information flow from different views, we propose trainable multi-view weights for each view as expressed in earlier equations. Our multi-view weights learn to assign larger weights to the view that comparatively contributes more in correct predication, and vice versa. For this, we use the cross-entropy loss of each view. Specifically, given ω_θ as the multi-view weights with $\theta = \{20°, 0°, -20°\}$, and loss of each view independently as $\mathcal{L}_\theta = -\sum y_i \ln \hat{y_{i,\theta}}$, we update the existing value of the multi-view weight as,

$$\omega_\theta := \frac{e^{-\sum y_i \ln \hat{y_{i,\theta}}}}{\sum_{j=1}^{K} e^{-\sum y_j \ln \hat{y_{j,\theta}}}}. \tag{9}$$

These learned weights reflect the contribution of each view, in terms of correct classification, and therefore, helps the recognition performance. Consequently, unlike the traditional approaches [36], [40], [41] which rely on constant and manually selected weights, we let the network learn the most suitable weights. This way, the networks not only learns to compute appropriate weights, but it also produces effective results which can be adapted for works where the need of manually-injected weights is discouraged [42], [43].

D. Faster Distributed Training

Contrary to the existing methods, we use PyTorch's distributed communication package (`torch.distributed`) for faster training capabilities, of multi-view learning, which are distributed over available resources. As a backend, we rely on NVIDIA Collective Communications Library (`NCCL`). By dynamically allocating available, and accessible, network's port and IP to form a TCP communication socket, we ensure synchronization of distributed processes in case of multiple GPUs or even multiple nodes. Our solution effectively offers communication primitives and support for multiprocess parallelism across several computation GPUs/nodes connected to one or more machines for efficient 3D/4D FER training.

TABLE I
ACCURACY (%) COMPARISONS WITH STATE-OF-THE-ART METHODS ON THE BU-3DFE SUBSET I AND SUBSET II, AND BOSPHORUS DATASETS

Method	Subset I
Zhen *et al.*[17]	84.50
Yang *et al.*[18]	84.80
Li *et al.*[19]	86.32
Li *et al.*[27]	86.86
Oyedotun *et al.*[28]	89.31
MiT - Tiny (Ours)	**91.57** (2.26↑)
MiT - Small (Ours))	**93.06** (3.75↑)
MiT - Base (Ours)	**94.32** (5.01↑)

Method	Subset II	Bosphorus
Li *et al.*[19]	80.42	79.72
Yang *et al.*[18]	80.46	77.50
Li *et al.*[27]	81.33	80.00
MiT - Tiny (Ours)	**85.79** (4.46↑)	**83.51** (3.51↑)
MiT - Small (Ours)	**87.13** (5.80↑)	**85.70** (5.70↑)
MiT - Base (Ours)	**88.25** (6.92↑)	**86.95** (6.95↑)

III. EXPERIMENTS AND RESULTS

To validate our proposed multi-view transformer architecture, we use Bosphorus [44], BU-3DFE [45], BU-4DFE [29] and BP4D-Spontaneous [46] datasets. Note that from the 3D/4D point cloud data, we first compute projected 2D images in multi-views by following previous works [36], [40], [41], [47]. Moreover, for video data, we form dynamic images via rank pooling [48] for each view [40] to feed our Transformer. For experiments, a 10-fold subject-independent cross-validation (CV) is used. Moreover, we validate our multi-view Transformer (MiT) by using the Tiny, Small and Base model variants - each with 14 patches, 12 layers and 224×224 resolution per view. The embedding dimension/heads/number of parameters in MiT-Tiny, MiT-Small and MiT-Base are 576/9/15 M, 1152/18/63 M, and 2304/36/255 M, respectively. This implies that the smaller MiT variant has a lower parameter count with faster throughput, and vice versa.

A. Comparisons With 3D FER Methods

As in the previous solutions [27], [28] for 3D FER, the BU-3DFE dataset containing 101 subjects is divided into two subsets: Subset I – the standard dataset including expressions with two higher levels of intensities, and Subset II – rarely applied in 3D FER, containing all four levels of intensities except the 100 neutral samples. For Bosphorus dataset, only 65 subjects perform the six expressions with each subject having one sample per expression. Table I summarizes the accuracy results from our extensive experiments for 3D FER. The overall impression from the reported results advocate the effectiveness of our method. Specifically, for Subset I dataset, we show that our multi-view transformer outperforms the previous most accurate state-of-the-art method [28] by 2.26%, 3.75% and 5.01%, when using the Tiny, Small and Base variants, respectively. A similar trend is noted for Subset II and Bosphorus datasets. From Tiny to Base, we improve the prediction results from 4.46% - 6.92% for Subset II, and from 3.51% - 6.95% for the Bosphorus dataset.

B. Comparisons With 4D FER Methods

We also conduct several experiments on BU-4DFE dataset which contains posed video clips of 101 subjects with six facial expressions. As shown in Table II, we report our accuracy results in comparison with several state-of-the-art methods. We illustrate that our model outperforms the competing methods by a considerable margin thanks to its effective multi-view strategy. Specifically, by comparing with the

TABLE II
PERFORMANCE (%) COMPARISON OF 4D FER WITH THE STATE-OF-THE-ART METHODS ON THE BU-4DFE DATASET

Method	Experimental Settings	Accuracy
Fang *et al.*[33]	10-CV, -	91.00
Li *et al.*[36]	10-CV, Full sequence	92.22
Ben Amor *et al.*[31]	10-CV, Full sequence	93.21
Zhen *et al.*[47]	10-CV, Full sequence	94.18
Bejaoui *et al.*[37]	10-CV, Full sequence	94.20
Zhen *et al.*[47]	10-CV, Key-frame	95.13
Behzad *et al.*[40]	10-CV, Full sequence	96.50
MiT - Tiny (Ours)*	10-CV, Full sequence	**97.23**(0.73↑)
MiT - Small (Ours)*	10-CV, Full sequence	**99.48**(2.98↑)
MiT - Base (Ours)*	10-CV, Full sequence	**99.66**(3.16↑)

*Click on Tiny, Small, and Base to access corresponding MiT.s training logs.

TABLE III
ACCURACY (%) COMPARISON ON THE BP4D-SPONTANEOUS DATASET. (A) RECOGNITION (B) CROSS-DATASET EVALUATION

(A)

Method	Accuracy
Yao *et al.*[49]	86.59
Danelakis *et al.*[50]	88.56
MiT - Base (Ours)	**91.67**(3.10↑)

(B)

Method	Accuracy
Zhang *et al.*[46]	71.00
Zhen *et al.*[47]	81.70
MiT - Base (Ours)	**84.03**(2.32↑)

most accurate state-of-the-art method [40], we demonstrate improvements of 0.73%, 2.98%, and 3.16%, by using the Tiny, Small and Base variants, respectively, reaching the highest accuracy of 99.66% on MiT-Base. This superior performance dictates that MiT offers a desirable solution via collaboration among embeddings from multi-views, together with our loss function.

C. Towards Spontaneous 4D FER

The BP4D-Spontaneous dataset contains a total of 41 subjects showing spontaneous expressions with two additional nervousness and pain expressions. In Table III, we report the recognition and cross-dataset evaluation results. With 3.10% improvement comparatively [50], MiT shows a dominant recognition performance by cruising to an accuracy of 91.67%. Following the experimental settings in [46], [47], we perform cross-dataset evaluations to highlight our model's robustness and generalizability. Here, the BU-4DFE dataset is chosen for training while a subset of the BP4D-Spontaneous dataset (i.e., Task 1 and Task 8, containing happy and disgust expressions) is chosen to validate the model's performance. With improvement of 2.32% compared to the most accurate method [47], MiT shows promising results indicating its robustness.

D. Multi-View Loss Ablation

We further demonstrate the effectiveness of the multi-view loss by comparing it with the loss of fusion of one ViT per view (as independent view) in Fig. 3. Comparatively, the closer accuracy gap in MiT-Small and MiT-Base against independent view is due to the higher complexity and number of parameters. Importantly, however, our multi-view loss is able to reduce the loss and boost the accuracy faster and to a better level. The is attributed to the gradient-friendly loss function and the trainable multi-view weights. We also show the role of each view towards

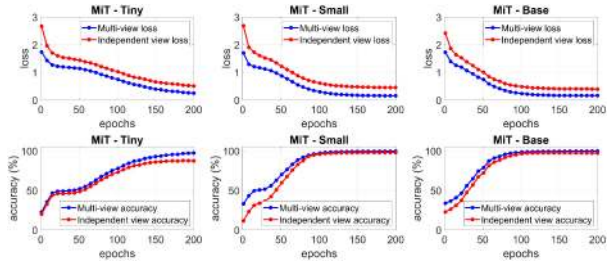

Fig. 3. Training loss and accuracy via proposed multi-view loss function.

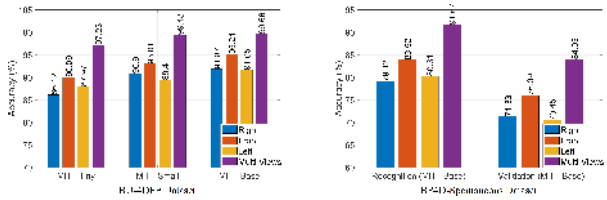

Fig. 4. Contribution of each view towards recognition.

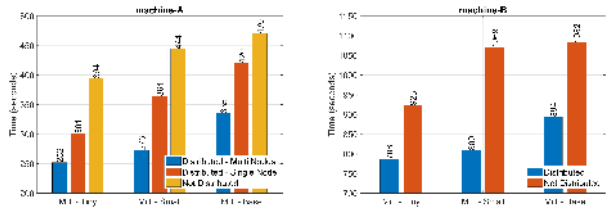

Fig. 5. Performance of distributed processing on different machines.

recognition in Fig. 4. Multi-view can achieve significantly better performance, i.e., of about 10%, than single views via the proposed MiT model.

E. Faster Distributed Training

We mainly use Xeon Gold 6230, NVIDIA Volta V100 GPUs (referred as machine-A) for experiments, but to demonstrate the performance efficiency via distributed training, we also report results for GP100GL, NVIDIA Tesla P100-PCIE GPUs (referred as machine-B) as well. In Fig. 5, we compare the average time to complete one epoch in training. As illustrated, the distributed processing yields results faster demonstrating the capability of our model to leverage available resources efficiently and effectively. The reported training times for Tiny, Small and Base models on machine-A are roughly 13.9 hours, 15.2 hours, and 18.6 hours, respectively.

IV. CONCLUSIONS

We presented MiT: a multi-view transformer model for 3D/4D facial affect recognition. MiT incorporates various facial patterns from multi-views and showed an effective recognition performance thanks additionally to our proposed multi-view loss function and trainable multi-view weights. Equipped with distributed training, our model showed significant results on widely-used datasets for 3D/4D FER.

ACKNOWLEDGMENT

The authors wish to acknowledge CSC – IT Center for Science, Finland, for the computational resources.

REFERENCES

[1] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://aclanthology.org/N19-1423/.

[3] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 121–125, 2021.

[4] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[5] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *CoRR*, 2021. [Online]. Available: https://arxiv.org/abs/2101.01169.

[6] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2020, pp. 5791–5800.

[7] M. Kumar, D. Weissenborn, and N. Kalchbrenner, "Colorization transformer," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=5NA1PinlGFu.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. IEEE Eur. Conf. Comput. Vision*, 2020, pp. 213–229.

[9] N. Parmar *et al.*, "Image transformer," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.

[10] M. Chen *et al.*, "Generative pretraining from pixels," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1691–1703.

[11] C. Liu, W. Zhou, Y. Chen, and J. Lei, "Asymmetric deeply fused network for detecting salient objects in RGB-D images," *IEEE Signal Process. Lett.*, vol. 27, pp. 1620–1624, 2020.

[12] R. Gao, F. Yang, W. Yang, and Q. Liao, "Margin loss: Making faces more separable," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 308–312, Feb. 2018.

[13] Y. Tian, J. Cheng, Y. Li, and S. Wang, "Secondary information aware facial expression recognition," *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1753–1757, Dec. 2019.

[14] P. Jiang, B. Wan, Q. Wang, and J. Wu, "Fast and efficient facial expression recognition using a gabor convolutional network," *IEEE Signal Process. Lett.*, vol. 27, pp. 1954–1958, 2020.

[15] M. Hu, Q. Chu, X. Wang, L. He, and F. Ren, "A two-stage spatiotemporal attention convolution network for continuous dimensional emotion recognition from facial video," *IEEE Signal Process. Lett.*, vol. 28, pp. 698–702, 2021.

[16] H. Li, J.-M. Morvan, and L. Chen, "3D facial expression recognition based on histograms of surface differential quantities," in *Proc. Int. Conf. Adv. Concepts Intell. Vision Syst*. 2011, pp. 483–494.

[17] Q. Zhen, D. Huang, Y. Wang, and L. Chen, "Muscular movement model-based automatic 3D/4D facial expression recognition," *IEEE Trans. Multimedia*, vol. 18, no. 7, pp. 1438–1450, Jul. 2016.

[18] X. Yang, D. Huang, Y. Wang, and L. Chen, "Automatic 3D facial expression recognition using geometric scattering representation," in *Proc. IEEE 11th Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2015, pp. 1–6.

[19] H. Li *et al.*, "An efficient multimodal 2D+ 3D feature-based approach to automatic facial expression recognition," *Comput. Vision Image Understanding*, vol. 140, pp. 83–92, 2015.

[20] H. Li *et al.* "3D facial expression recognition via multiple kernel learning of multi-scale local normal patterns," in *Proc. 21st Int. Conf. Pattern Recognit.*, 2012, pp. 2577–2580.

[21] X. Li, T. Jia, and H. Zhang, "Expression-insensitive 3D face recognition using sparse representation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 2575–2582.

[22] O. Ocegueda, T. Fang, S. K. Shah, and I. A. Kakadiaris, "Expressive maps for 3D facial expression recognition," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 1270–1275.

[23] I. Mpiperis, S. Malassiotis, and M. G. Strintzis, "Bilinear models for 3-D face and facial expression recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 3, pp. 498–511, Sep. 2008.

[24] X. Zhao, D. Huang, E. Dellandréa, and L. Chen, "Automatic 3D facial expression recognition based on a bayesian belief net and a statistical facial feature model," in *Proc. Int. Conf. Pattern Recognit.*, 2010, pp. 3724–3727.

[25] C. Samir *et al.* "An intrinsic framework for analysis of facial surfaces," *Int. J. Comput. Vision*, vol. 82, no. 1, pp. 80–95, 2009.

[26] A. Maalej, B. B. Amor, M. Daoudi, A. Srivastava, and S. Berretti, "Shape analysis of local facial patches for 3D facial expression recognition," *Pattern Recognit.*, vol. 44, no. 8, pp. 1581–1589, 2011.

[27] H. Li, J. Sun, Z. Xu, and L. Chen, "Multimodal 2D+ 3D facial expression recognition with deep fusion convolutional neural network," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2816–2831, Dec. 2017.

[28] O. K. Oyedotun, G. Demisse, A. E. R. Shabayek, D. Aouada, and B. Ottersten, "Facial expression recognition via joint deep learning of RGB-depth map latent representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vision Workshops*, 2017, pp. 3161–3168.

[29] X. Zhang *et al.*, "A high-resolution spontaneous 3D dynamic facial expression database," in *Proc. 10th Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2013, pp. 1–6.

[30] Y. Sun, X. Chen, M. Rosato, and L. Yin, "Tracking vertex flow and model adaptation for three-dimensional spatiotemporal face analysis," *IEEE Trans. Syst., Man, Cybern. A, Syst. Hum.*, vol. 40, no. 3, pp. 461–474, May 2010.

[31] B. B. Amor, H. Drira, S. Berretti, M. Daoudi, and A. Srivastava, "4-D facial expression recognition by learning geometric deformations," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2443–2457, Dec. 2014.

[32] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert, "Recognition of 3D facial expression dynamics," *Image Vision Comput.*, vol. 30, no. 10, pp. 762–773, 2012.

[33] T. Fang, X. Zhao, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris, "3D/4D facial expression analysis: An advanced annotated face model approach," *Image Vision Comput.*, vol. 30, no. 10, pp. 738–749, 2012.

[34] T. Fang, X. Zhao, S. K. Shah, and I. A.Kakadiaris, "4D facial expression recognition," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 1594–1601.

[35] M. Reale, X. Zhang, and L. Yin, "Nebula feature: A space-time feature for posed and spontaneous 4D facial behavior analysis," in *Proc. IEEE 10th Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2013, pp. 1–8.

[36] W. Li, D. Huang, H. Li, and Y. Wang, "Automatic 4D facial expression recognition using dynamic geometrical image network," in *Proc. IEEE 13th Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2018, pp. 24–30.

[37] H. Bejaoui, H. Ghazouani, and W. Barhoumi, "Sparse coding-based representation of LBP difference for 3D/4D facial expression recognition," *Multimedia Tools Appl.*, vol. 78, no. 16, pp. 22 773–22 796, 2019.

[38] P. D. Marrero Fernandez, F. A. Guerrero Pena, T. Ren, and A. Cunha, "Feratt: Facial expression recognition with attention net," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, pp. 837–846, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9025630.

[39] F. Ma, B. Sun, and S. Li, "Robust facial expression recognition with convolutional visual transformers," 2021. [Online]. Available: https://arxiv.org/abs/2103.16854.

[40] M. Behzad, N. Vo, X. Li, and G. Zhao, "Automatic 4D facial expression recognition via collaborative cross-domain dynamic image network," in *Brit. Mach. Vision Conf.*, 2019.

[41] M. Behzad, N. Vo, X. Li, and G. Zhao, "Towards reading beyond faces for sparsity-aware 3D/4D affect recognition," *Neurocomputing*, vol. 458, pp. 297–307, 2021.

[42] M. A. Mahmoudi, A. Chetouani, F. Boufera, and H. Tabia, "Learnable pooling weights for facial expression recognition," *Pattern Recognit. Lett.*, vol. 138, pp. 644–650, 2020.

[43] B. Yang, J. Cao, R. Ni, and Y. Zhang, "Facial expression recognition using weighted mixture deep neural network based on double-channel facial images," *IEEE Access*, vol. 6, pp. 4630–4640, 2018.

[44] A. Savran *et al.*, "Bosphorus database for 3D face analysis," in *Proc. Eur. Workshop Biometrics Identity Manage.*, 2008, pp. 47–56.

[45] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proc. IEEE 7th Int. Conf. Workshops Autom. Face Gesture Recognit.*, 2006, pp. 211–216.

[46] X. Zhang *et al.*, "BP4D-spontaneous: A high-resolution spontaneous 3D dynamic facial expression database," *Image Vision Comput.*, vol. 32, no. 10, pp. 692–706, 2014.

[47] Q. Zhen, D. Huang, H. Drira, B. B. Amor, Y. Wang, and M. Daoudi, "Magnifying subtle facial motions for effective 4D expression recognition," *IEEE Trans. Affect. Comput.*, vol. 10, no. 4, pp. 524–536, Oct.–Dec. 2019.

[48] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2799–2813, Dec. 2018.

[49] Y. Yao, D. Huang, X. Yang, Y. Wang, and L. Chen, "Texture and geometry scattering representation-based facial expression recognition in 2D+3D videos," *ACM Trans. Mult. Comput. Commun. Appl.*, vol. 14, pp. 1–23, 2018.

[50] A. Danelakis, T. Theoharis, I. Pratikakis, and P. Perakis, "An effective methodology for dynamic 3D facial expression retrieval," *Pattern Recognit.*, vol. 52, pp. 174–185, 2016.